DSP 7600: Assignment 2

Prepared by Andrew Stone (201534971)

Prepared for Dr. Sarah Power

Section 1 |

1.1 |

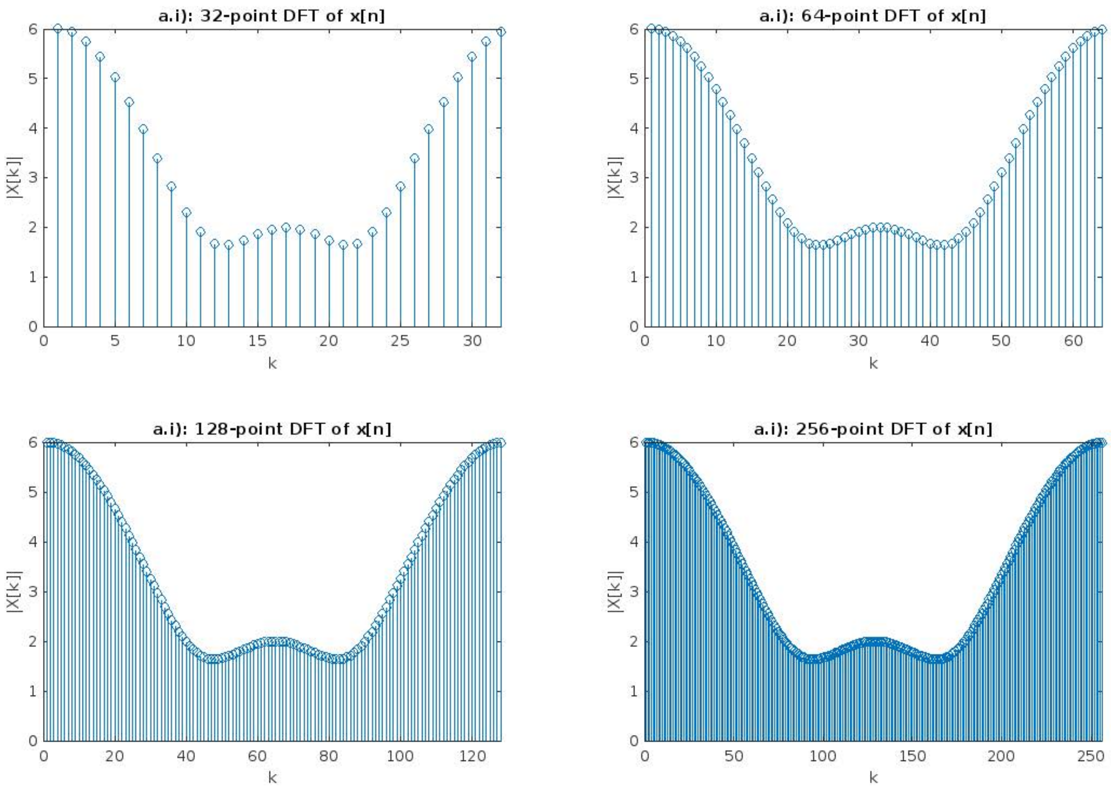
$$\text{i)}\quad x[n] = \begin{cases} \begin{cases} 3 & n=0 \\ 2 & n=1 \end{cases} \\ \begin{cases} 1 & n=2 \\ 0 & \text{else} \end{cases} \end{cases} \qquad \text{ii)}\quad x[n] = \begin{cases} 1 & 0 \le n \le 6 \\ 0 & \text{else} \end{cases}$$

1.a | DFT

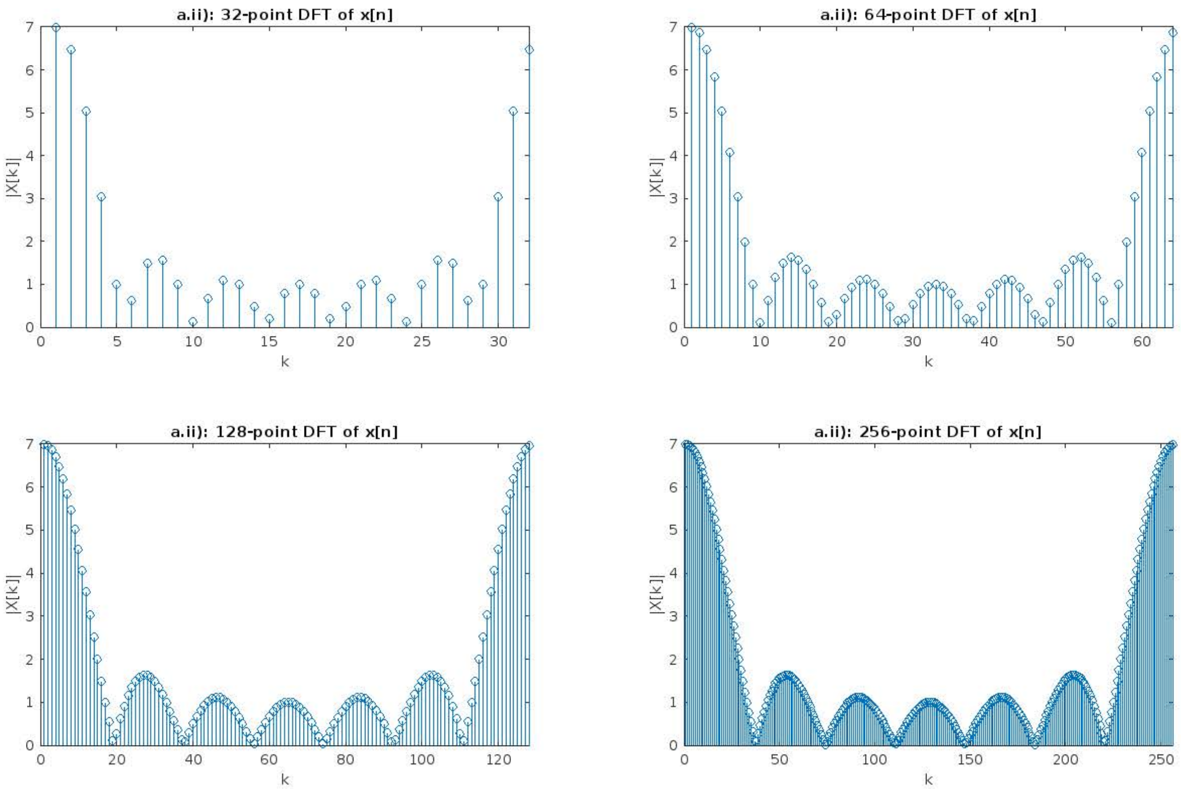
Compute N-Point DFT via transform equation for n ∈ N = { 32, 64, 128, 256 }

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\omega_o n} = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}\text{kn}}, \; k = 0, 1, 2, \ldots, N-1$$

```
% i)
xi = [3 2 1];
Xk_32 = DFT(xi, 32);
Xk_64 = DFT(xi, 64);
Xk_128 = DFT(xi, 128);
Xk_256 = DFT(xi, 256);

% figure1 = figure("Name", "N-point DFT", "Color", "black");
figure1 = figure("Name", "N-point DFT");

List = { Xk_32, Xk_64, Xk_128, Xk_256 };
ListName = [ "32-point", "64-point", "128-point", "256-point" ];

for i = 1 : length(List)
    subplot(2,2, i);
    stem(abs( cell2mat(List(i)) ));
    xlabel('k');
    xlim( [0 length( cell2mat(List(i)) )] );
    ylabel('|X[k]|');
    title( sprintf("a.i: %s DFT of x[n]", ListName(i)) );
end
```




```
% ii)
xii = [1 1 1 1 1 1 1];
Xk_32 = DFT(xii, 32);
Xk_64 = DFT(xii, 64);
Xk_128 = DFT(xii, 128);
Xk_256 = DFT(xii, 256);

figure2 = figure("Name","N-point DFT");

List = { Xk_32, Xk_64, Xk_128, Xk_256 };
ListName = [ "32-point", "64-point", "128-point", "256-point" ];

for i = 1 : length(List)
    subplot(2,2, i);
    stem(abs( cell2mat(List(i)) ));
    xlabel('k');
    xlim( [0 length( cell2mat(List(i)) )] );
    ylabel('|X[k]|');
    title( sprintf("a.ii: %s DFT of x[n]", ListName(i)) );
end
```



1.b | DTFT

$$x_i[n] = \begin{cases} \begin{cases} 3 & n=0 \\ 2 & n=1 \end{cases} \\ \begin{cases} 1 & n=2 \\ 0 & \text{else} \end{cases} \end{cases}$$

$$X_i\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} = \sum_{n=0}^{2} x[n]e^{-j\omega n} = \sum 3e^{0} + 2e^{-j\,\omega} + e^{-j\,2\omega}$$

$$= 3 + 2e^{-j\,\omega} + e^{-j\,2\omega}$$

$X_i\left(e^{j\omega}\right)$ evaluated at $\omega = \dfrac{2\pi}{32}$ and $\omega = \dfrac{9\pi}{8}\ldots$

$$X_i\left(e^{j\frac{2\pi}{32}}\right) = 5.9360\ ;\quad X_{\text{ii}}\left(e^{j\frac{9\pi}{8}}\right) = 1.8603$$

Corrisponding k value for various N such that $X[n] = 5.9360$

$$N = 32 \rightarrow k = \frac{2\pi}{32} \div \frac{2\pi}{32} = 1 \rightarrow X_i[1] = 5.9360$$

$$N = 64 \rightarrow k = \frac{2\pi}{32} \div \frac{2\pi}{64} = 2 \rightarrow X_i[2] = 5.9360$$

$$N = 128 \rightarrow k = \frac{2\pi}{32} \div \frac{2\pi}{128} = 4 \rightarrow X_i[4] = 5.9360$$

$$N = 256 \rightarrow k = \frac{2\pi}{32} \div \frac{2\pi}{256} = 8 \rightarrow X_i[8] = 5.9360$$

Corrisponding *k* value for various N such that $X[n] = 1.8603$

$$N = 32 \rightarrow k = \frac{9\pi}{8} \div \frac{2\pi}{32} = 18 \rightarrow X_i[18] = 1.8603$$

$$N = 64 \rightarrow k = \frac{9\pi}{8} \div \frac{2\pi}{64} = 36 \rightarrow X_i[36] = 1.8603$$

$$N = 128 \rightarrow k = \frac{9\pi}{8} \div \frac{2\pi}{128} = 72 \rightarrow X_i[72] = 1.8603$$

$$N = 256 \rightarrow k = \frac{9\pi}{8} \div \frac{2\pi}{256} = 144 \rightarrow X_i[144] = 1.8603$$

$$x_{\text{ii}}[n] = \begin{cases} 1 & 0 \leq n \leq 6 \\ 0 & \text{else} \end{cases}$$

$$X_{\text{ii}}\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} = \sum_{n=0}^{6} x[n]e^{-j\omega n} = \sum (1)e^{0} + (1)e^{-j\,\omega} + (1)e^{-j\,2\omega} + (1)e^{-j\,3\omega} + (1)e^{-j\,4\omega} + (1)e^{-j\,5\omega} + (1)e^{-j\,6\omega}$$

$$= 1 + e^{-j\,\omega} + e^{-j\,2\omega} + e^{-j\,3\omega} + e^{-j\,4\omega} + e^{-j\,5\omega} + e^{-j\,6\omega}$$

$X_{\text{ii}}\left(e^{j\omega}\right)$ evaluated at $\omega = \frac{2\pi}{32}$ and $\omega = \frac{9\pi}{8}\ldots$

$$X_{\text{ii}}\left(e^{j\frac{2\pi}{32}}\right) = 6.4723\ ; \quad X_{\text{ii}}\left(e^{j\frac{9\pi}{8}}\right) = 0.1989$$

Corrisponding *k* value for various N such that $X[n] = 6.4723$

$$N = 32 \rightarrow k = \frac{2\pi}{32} \div \frac{2\pi}{32} = 1 \rightarrow X_{\text{ii}}[1] = 6.4723$$

$$N = 64 \rightarrow k = \frac{2\pi}{32} \div \frac{2\pi}{64} = 2 \rightarrow X_{\text{ii}}[2] = 6.4723$$

$$N = 128 \rightarrow k = \frac{2\pi}{32} \div \frac{2\pi}{128} = 4 \rightarrow X_{\text{ii}}[4] = 6.4723$$

$$N = 256 \rightarrow k = \frac{2\pi}{32} \div \frac{2\pi}{256} = 8 \rightarrow X_{\text{ii}}[8] = 6.4723$$

Corrisponding *k* value for various N such that $X[n] = 0.1989$

$$N = 32 \rightarrow k = \frac{9\pi}{8} \div \frac{2\pi}{32} = 18 \rightarrow X_{\text{ii}}[18] = 0.1989$$

$$N = 64 \rightarrow k = \frac{9\pi}{8} \div \frac{2\pi}{64} = 36 \rightarrow X_{\text{ii}}[36] = 0.1989$$

$$N = 128 \rightarrow k = \frac{9\pi}{8} \div \frac{2\pi}{128} = 72 \rightarrow X_{\text{ii}}[72] = 0.1989$$

$$N = 256 \rightarrow k = \frac{9\pi}{8} \div \frac{2\pi}{256} = 144 \rightarrow X_{\text{ii}}[144] = 0.1989$$

Comments:

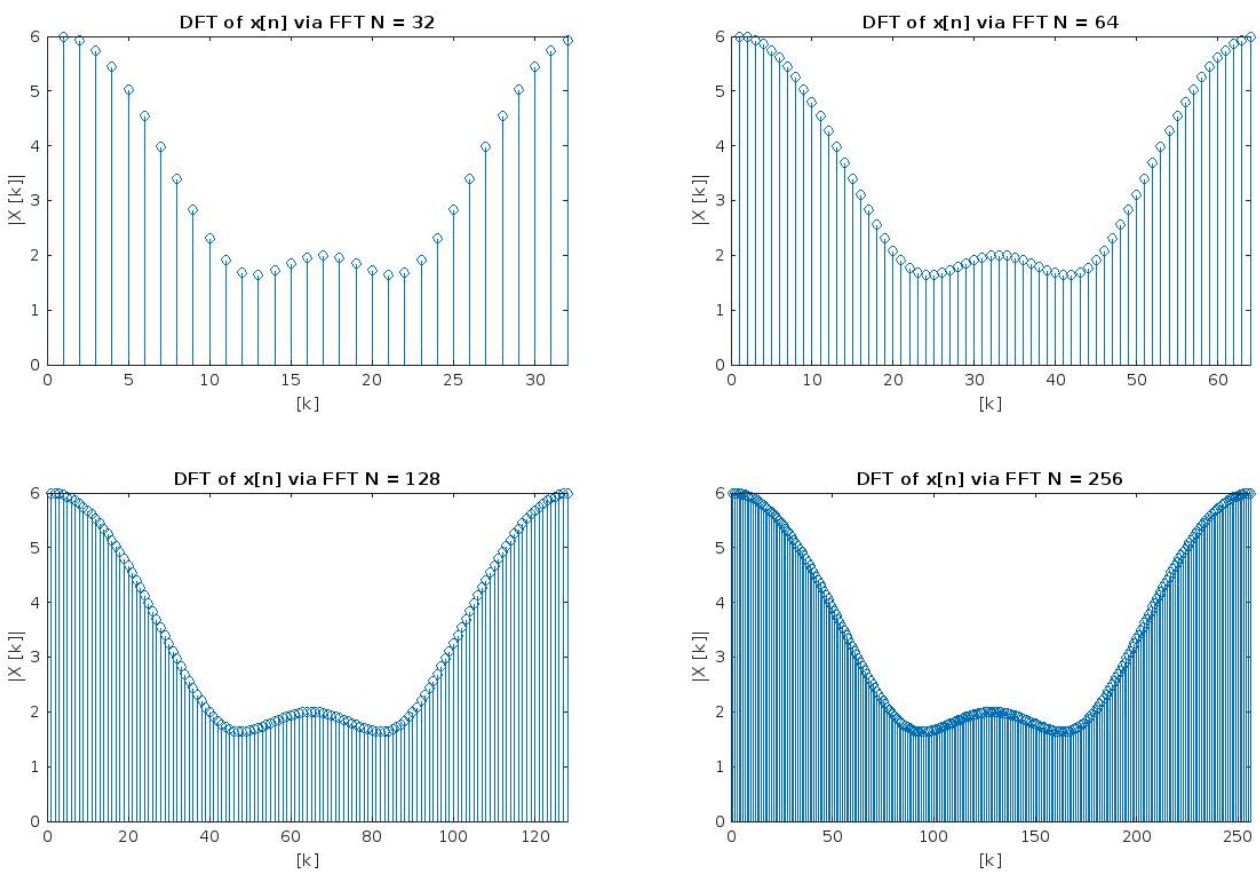
- Magnitude of DFT are the same for all *k* corrisponding to ω.
- This is a result of the evaluation of $\frac{2\pi}{N}$ which dictates the "distance" between subsequent frequency values.
- Because $\frac{2\pi}{32}$ and $\frac{9\pi}{8}$ is an integer multiple of $\frac{2\pi}{N}$, there exists integer solutions to $\omega = k\,\frac{2\pi}{N}$.

```
function out = DFT(x_, N_)
    x = [x_ zeros(1, (N_-length(x_)))]; % padd input array x_ to match length of N_
    out = zeros(1, N_);
    for k = 0 : N_-1
        for n = 0 : N_-1
            out(k+1) = out(k+1) + x(n+1) * exp(1j * 2 * pi() / N_ * k * n);
        end
    end
end
```

Section 2 |

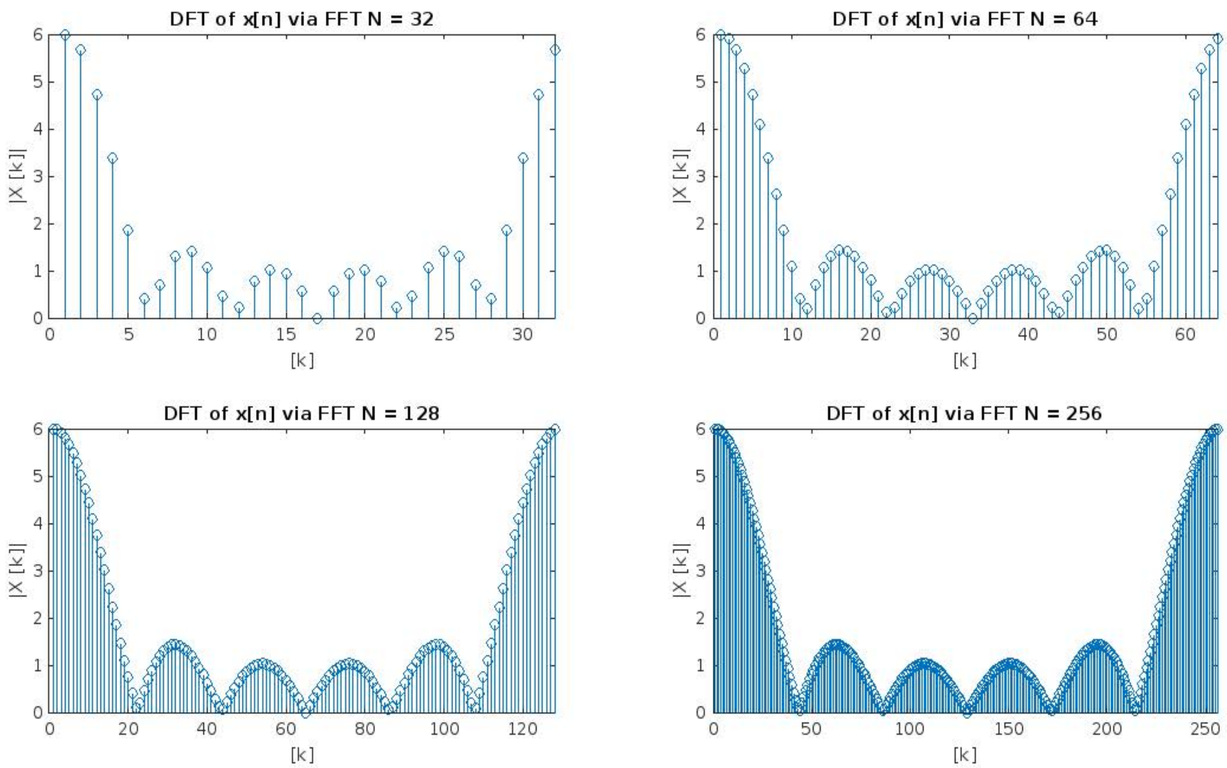
$$x[n] = \begin{cases} \begin{cases} 3 & n = 0 \\ 2 & n = 1 \end{cases} \\ \begin{cases} 1 & n = 2 \\ 0 & \text{else} \end{cases} \end{cases}$$

```
xi = [3 2 1];
ListSizeN = [ 32, 64, 128, 256 ];
ListName = [ "N = 32", "N = 64", "N = 128", "N = 256" ];

figure1 = figure("Name","DFT via FFT");

for i = 1 : length(ListSizeN)
    subplot(2,2, i);
    xi_ = [xi zeros(1,( ListSizeN(i)-length(xi) ) ) ];
    fft_ = fft(xi_);
    stem( abs( fft_ ) );
    xlabel('[k]');
    xlim( [0, ListSizeN(i)] );
    ylabel('|X [k]|');
    title( sprintf("DFT of x[n] via FFT %s", ListName(i)) );
end
```



$$x[n] = \begin{cases} 1 & 0 \le n \le 6 \\ 0 & \text{else} \end{cases}$$

```
xii = [1, 1, 1, 1, 1, 1];
figure2 = figure("Name","DFT via FFT");

for i = 1 : length(ListSizeN)
    subplot(2,2, i);
    xii_ = [xii zeros(1,( ListSizeN(i)-length(xii) ) ) ];
    fft_ = fft(xii_);
    stem( abs( fft_ ) );
    xlabel('[k]');
    xlim( [0, ListSizeN(i)] );
    ylabel('|X [k]|');
    title( sprintf("DFT of x[n] via FFT %s", ListName(i)) );
end
```

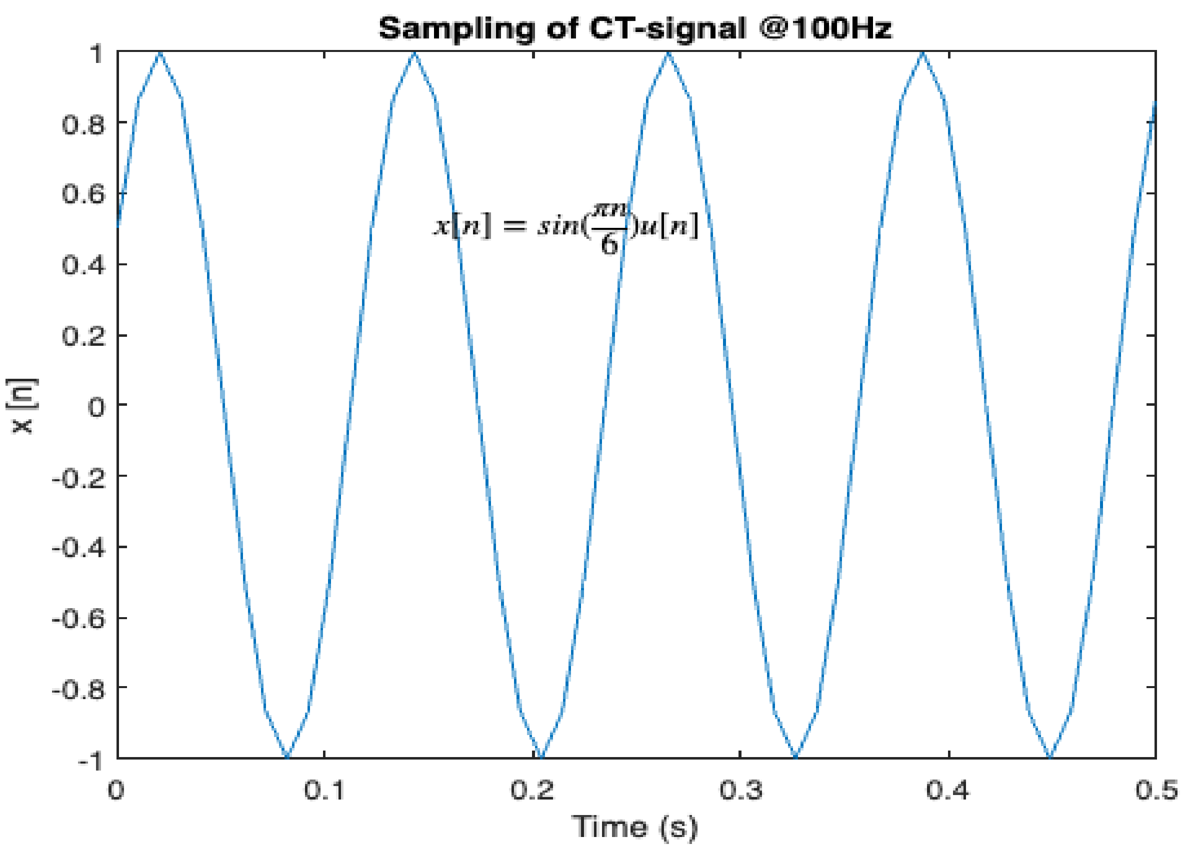


Comments:

- In both cases of $x[n]$, the results of using Fast Fourier Transform match the results obtained in section 1 by computing the DFT via transform equation.

Section 3 |

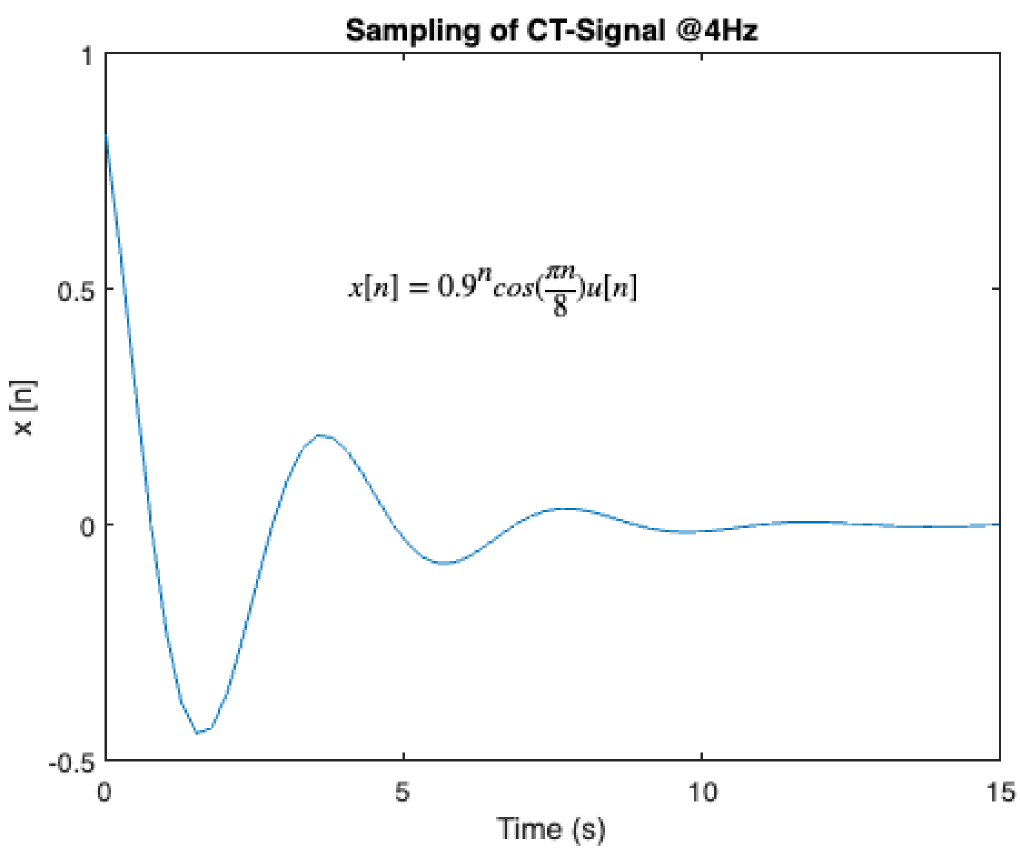
```
% 3a)
t_start = 0;
t_end = 0.5;
sampleFrequency = 100;
totalSamplePoints = (t_end - t_start) * sampleFrequency;

DTtime = linspace(t_start, t_end, totalSamplePoints);
xn = zeros(1, length(DTtime));

for n = 1 : length(DTtime)
    xn(n) = sin(pi/6 * n);
end

figure1 = figure("Name", "DT Sampling of CT-signal @100Hz");
plot(DTtime, xn);
title("Sampling of CT-signal @100Hz");
text(0.15, 0.5, "$$x[n]=sin(\frac{\pi n}{6})u[n]$$", Interpreter="latex");
xlabel('Time (s)');
ylabel('x [n]');
```



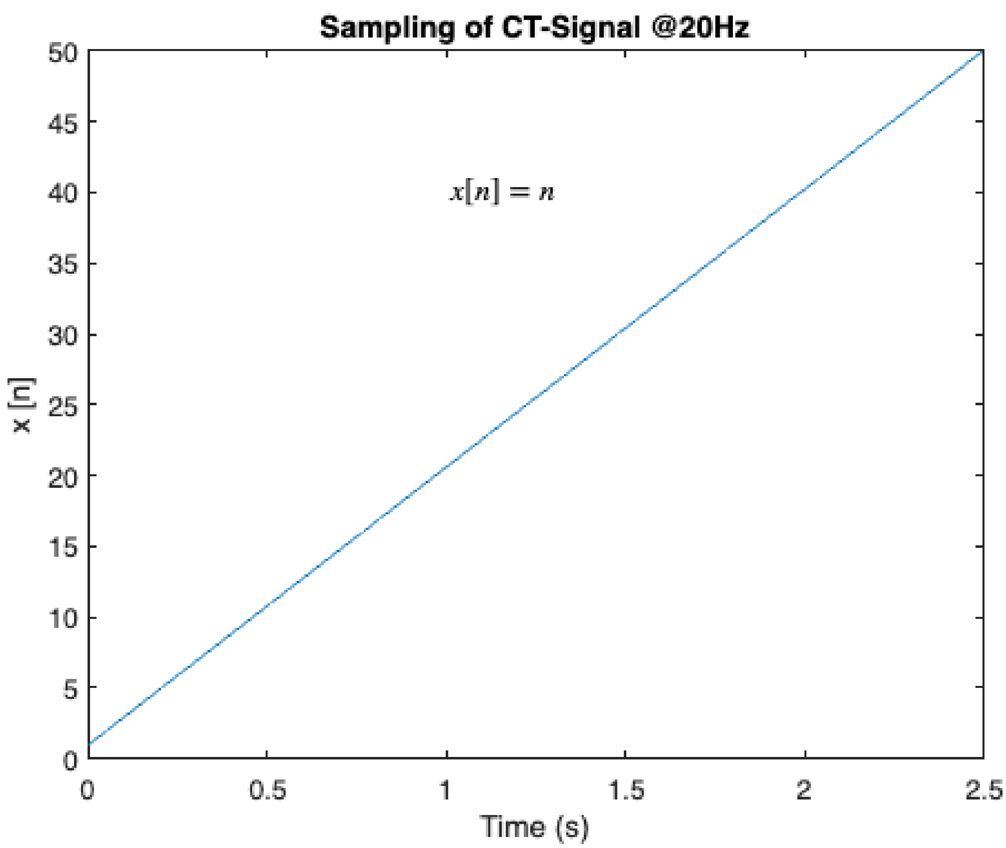
```
% 3b)
t_start = 0;
t_end = 15;
```

```
sampleFrequency = 1/0.25;
totalSamplePoints = (t_end - t_start) * sampleFrequency;
DTtime = linspace(t_start, t_end, totalSamplePoints);
xn = zeros(1, length(DTtime));

for n = 1 : length(DTtime)
    xn(n) = 0.9^n * cos(pi/8 * n);
end
```

```
figure2 = figure("Name", "DT Sampling of CT-Signal @4Hz");
plot(DTtime, xn);
title("Sampling of CT-Signal @4Hz");
text(4, 0.5, "$$x[n]=0.9^ncos(\frac{\pi n}{8})u[n]$$", Interpreter="latex");
xlabel("Time (s)");
ylabel("x [n]");
```



```
% 3c)
t_start = 0;
t_end = 2.5;
sampleFrequency = 20;
totalSamplePoints = (t_end - t_start) * sampleFrequency;
DTtime = linspace(t_start, t_end, totalSamplePoints);
xn = zeros(1, length(DTtime));

for n = 1 : length(DTtime)
    xn(n) = n;
end

figure3 = figure("Name", "DT Sampling of CT-Signal @20Hz");
plot(DTtime, xn);
title("Sampling of CT-Signal @20Hz");
text(1, 40, "$$x[n]=n$$", Interpreter="latex");
xlabel("Time (s)");
ylabel("x [n]");
```

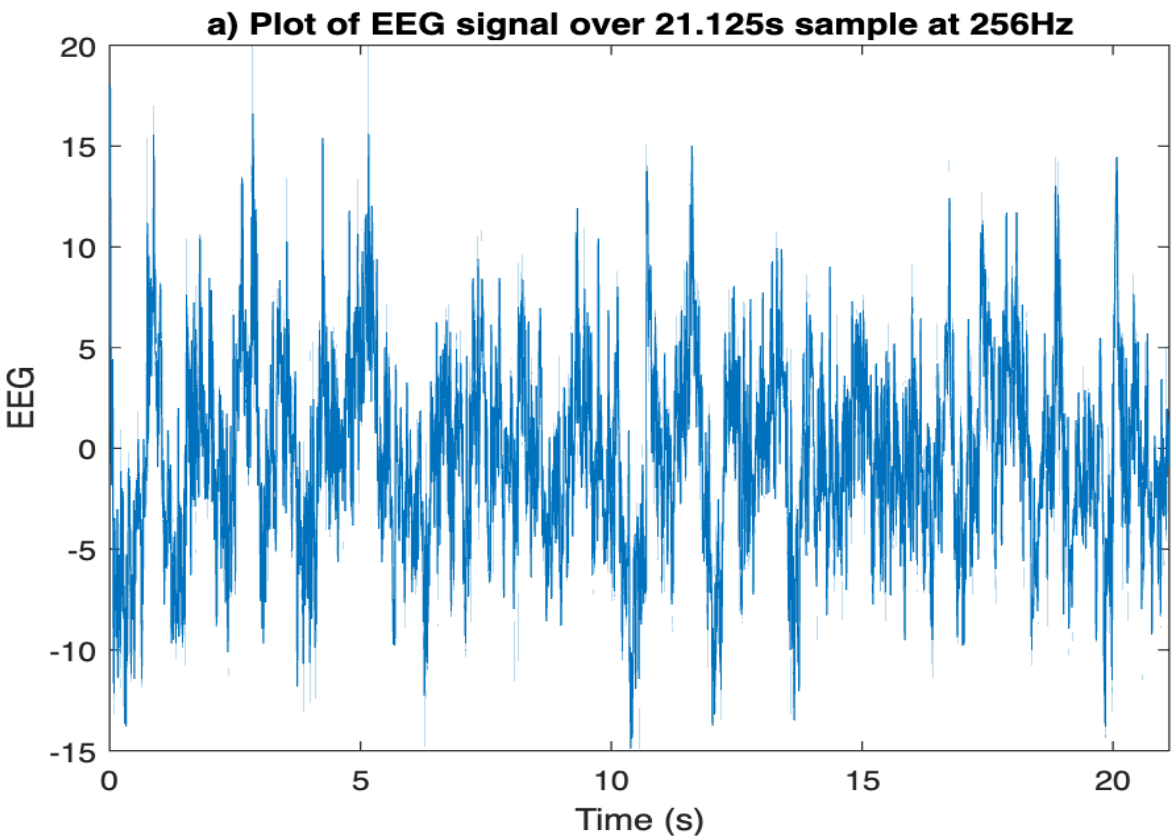


Section 4 |

```
% a)
EEG=load('Assign2_eeg.mat');
[L, tmp] = size(EEG.data);
T = (L-1)/256;
t = 0 : 1/256 : T;
w = 0 : 2*pi / (21.125*256) : 2*pi();
f = 0 : 256/(21.125*256) : 256;
```

Signal duration is $(5409-1)/256\text{Hz} = 21.125\ \text{sec}$

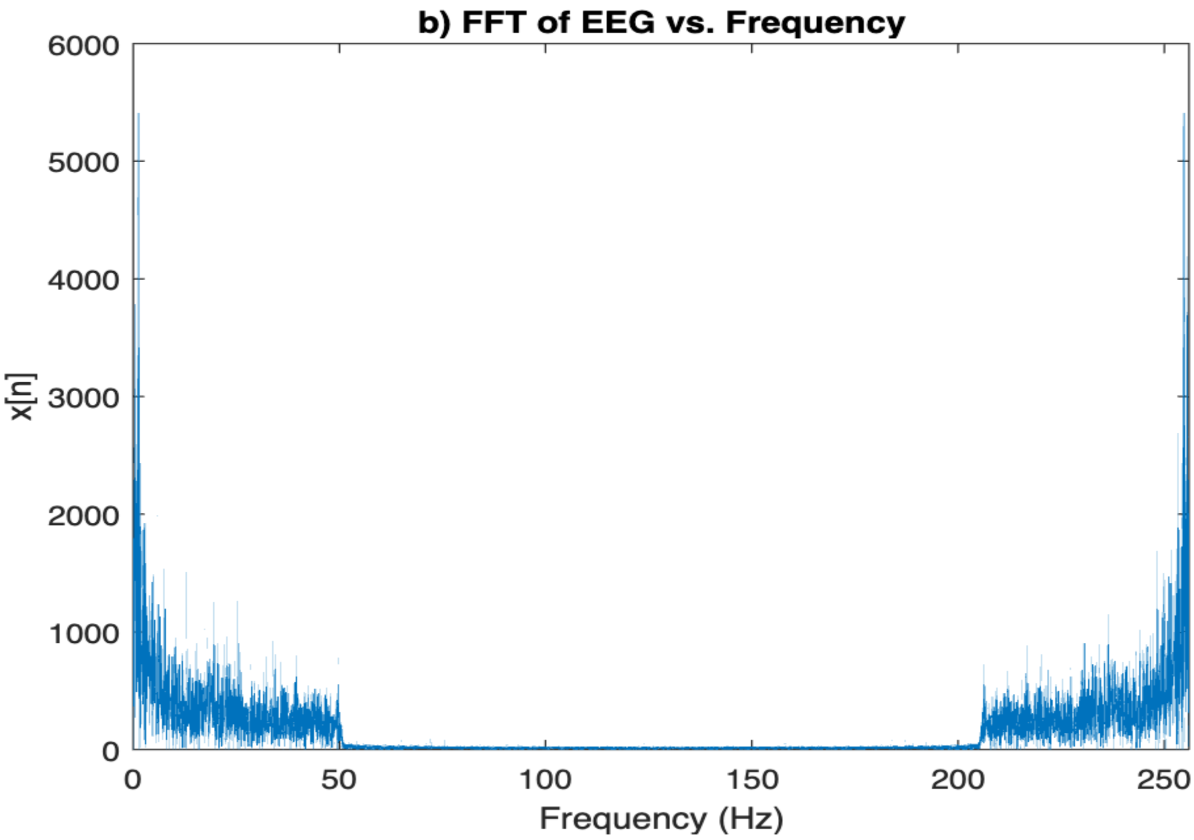
```
figure1 = figure();
plot(t, EEG.data);
title('a) Plot of EEG signal over 21.125s sample at 256Hz');
xlim([0, T])
xlabel('Time (s)');
ylabel('EEG');
```




```
% b)
```

The cutoff frequency of the low pass filter can be determined by seeing what frequency components have been suppressed in the FFT representation of the signal. It is clear from the following figure that the FFT is suppressed at frequencies higher than 50Hz, the higher frequencies are really a mirror of the lower frequencies, and they correspond to the lower same DT frequencies as the lower negative frequencies.

```
figure2 = figure();
plot(f, abs(fft(EEG.data)));
xlim([0, max(f)])
title('b) FFT of EEG vs. Frequency');
xlabel('Frequency (Hz)');
ylabel('x[n]');
```



$$f_c = 50\,\text{Hz}$$

$$f_s = 256\,\text{Hz} = \frac{1}{T_s}$$

$$T_s = \frac{1}{256}s$$

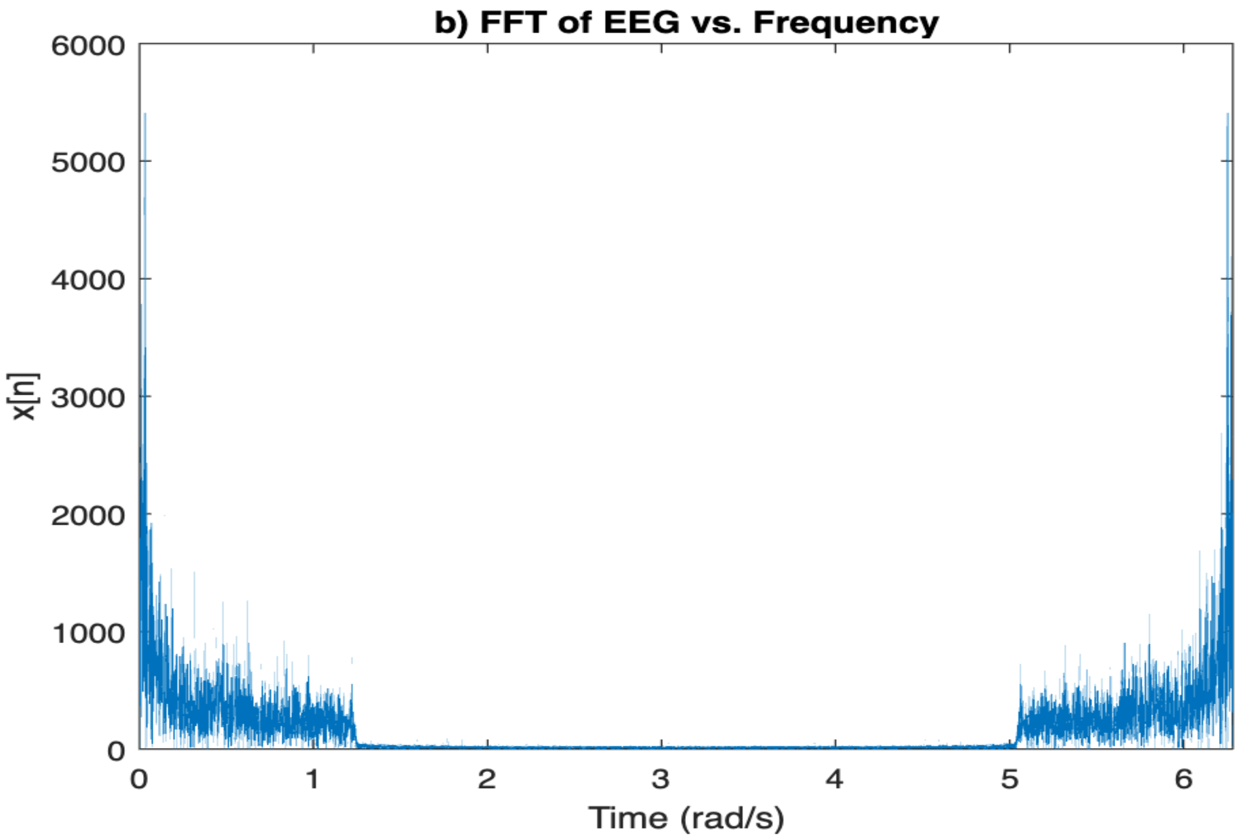
$$\Omega_c = 2\pi f_c = 100\pi$$

$$\omega_c = \Omega_c T_s = 100\pi * \frac{1}{256} = 1.227$$

1.227 is shown in the next graph to shows the cutoff frequency in rads/sample

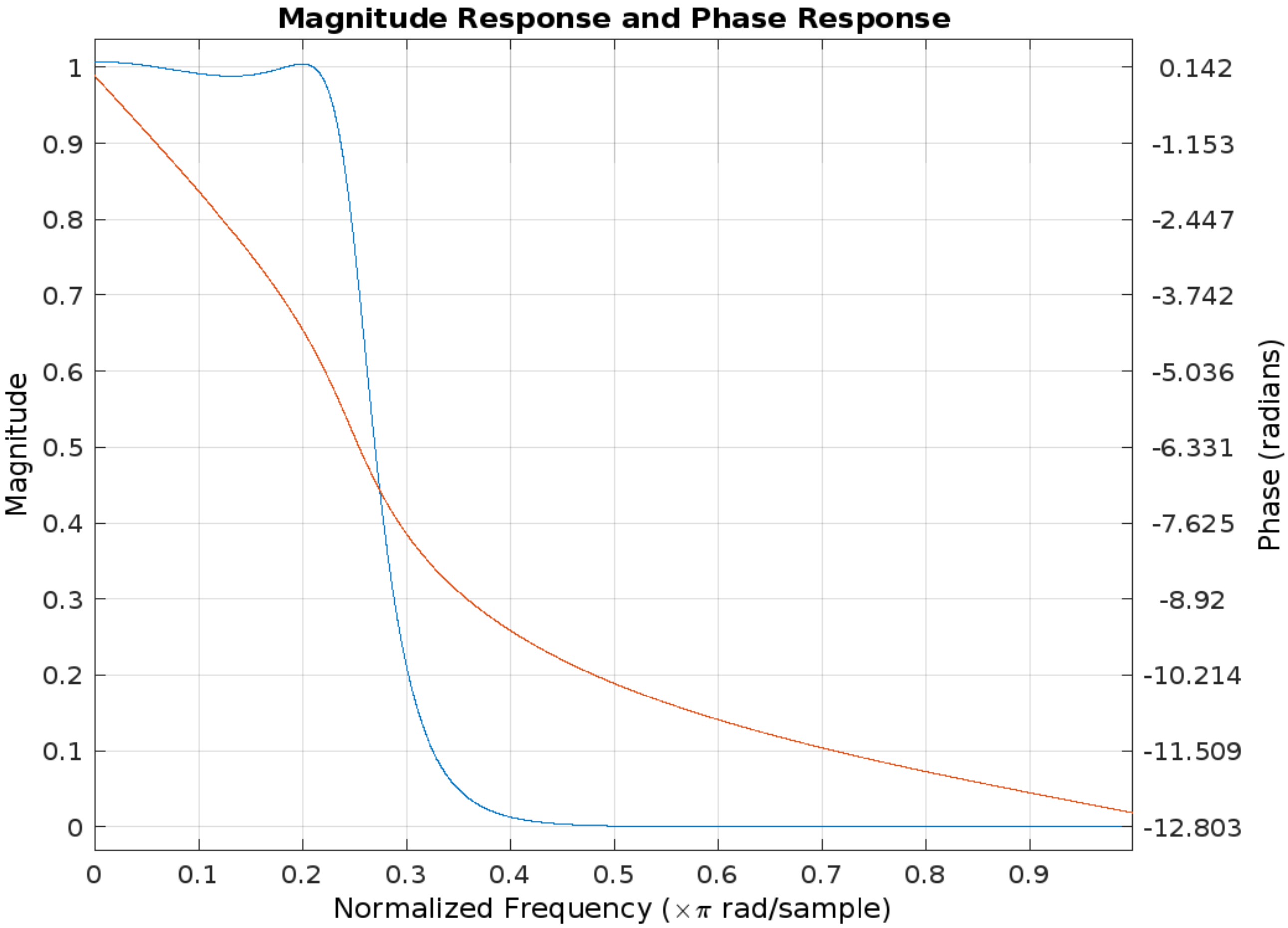
```
figure3 = figure();
```

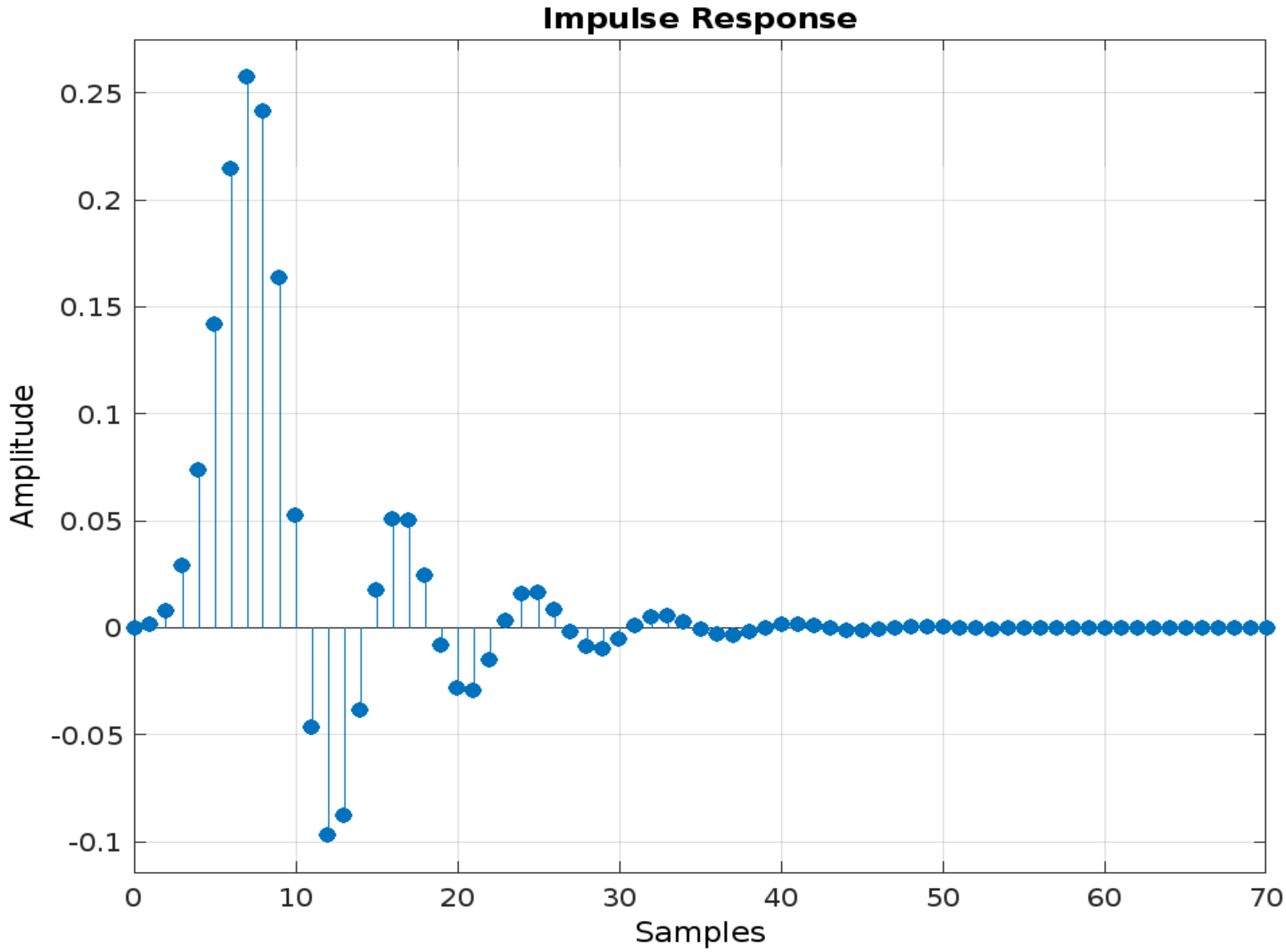
```
plot(w, abs(fft(EEG.data)));
xlim([0 max(w)])
title('b) FFT of EEG vs. Frequency');
xlabel('Time (rad/s)');
ylabel('x[n]');
```

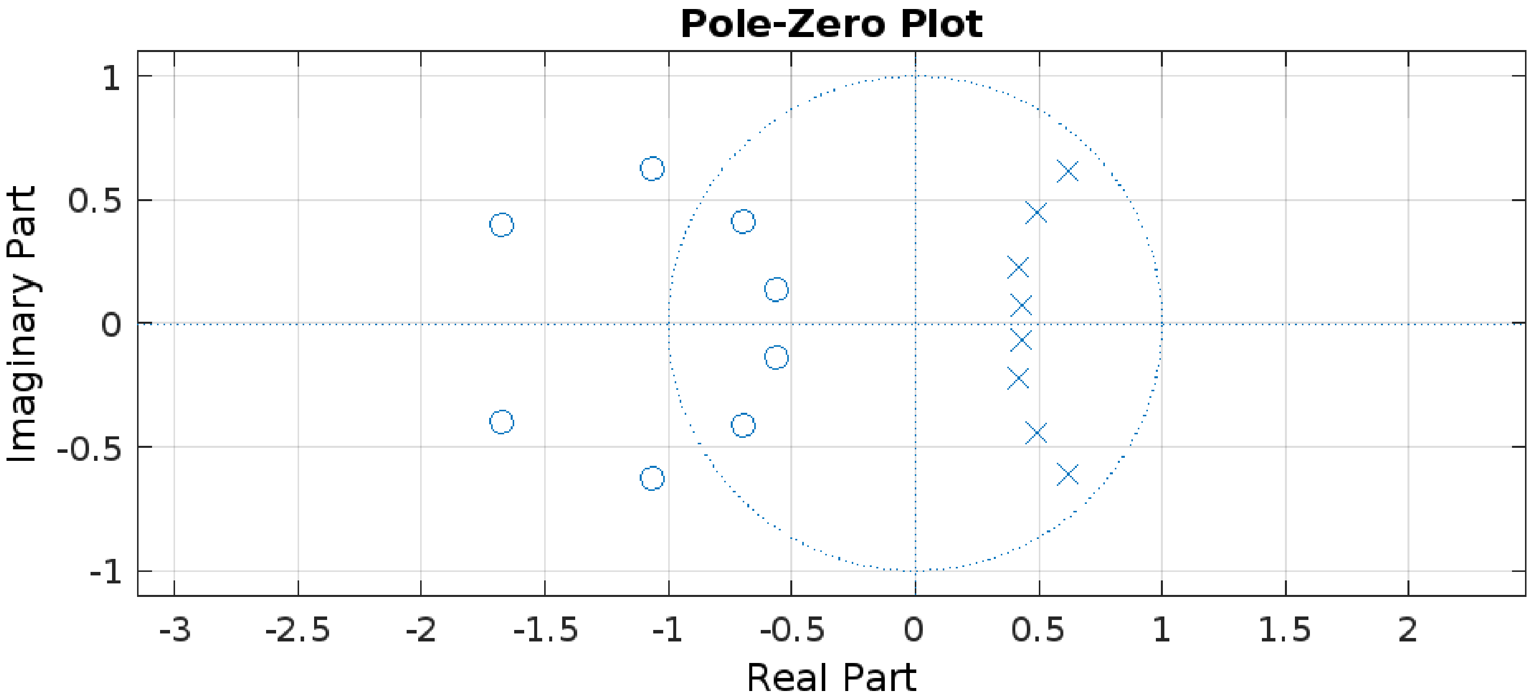


Section 5

```
% a)

% 0.0001201z^8 + 0.0009608z^7 + 0.003363z^6 + 0.006725z^5 + 0.008407z^4 + 0.006725z^3 +
% ----------------------------------------------------------------------------------------
% z^8 - 3.919z^7 + 7.325z^6 - 8.275z^5 + 6.106z^4 - 2.989z^3 + 0.9423z^2 - 0.1742z + 0.
%
% num = [0.0001201 + 0.0009608 + 0.003363 + 0.006725 + 0.008407 + 0.006725 + 0.003363 +
% denom = [ 1 + 3.919 + 7.325 - 8.275 + 6.106 - 2.989 + 0.9423 - 0.1742 + 0.01442];

% fvtool(0.0001201, 0.0009608, 0.003363, 0.006725, 0.008407, 0.006725, 0.003363, 0.0009


num = [0.0001201, 0.0009608, 0.003363, 0.006725, 0.008407, 0.006725, 0.003363, 0.00096
denom = [1, -3.919, 7.325, -8.275, 6.106, -2.989, 0.9423, -0.1742, 0.01442];

% fvtool(num, denom);
```




Impulse Response
Amplitude
Samples
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
0
10
20
30
40
50
60
70



Section 5 | part b)

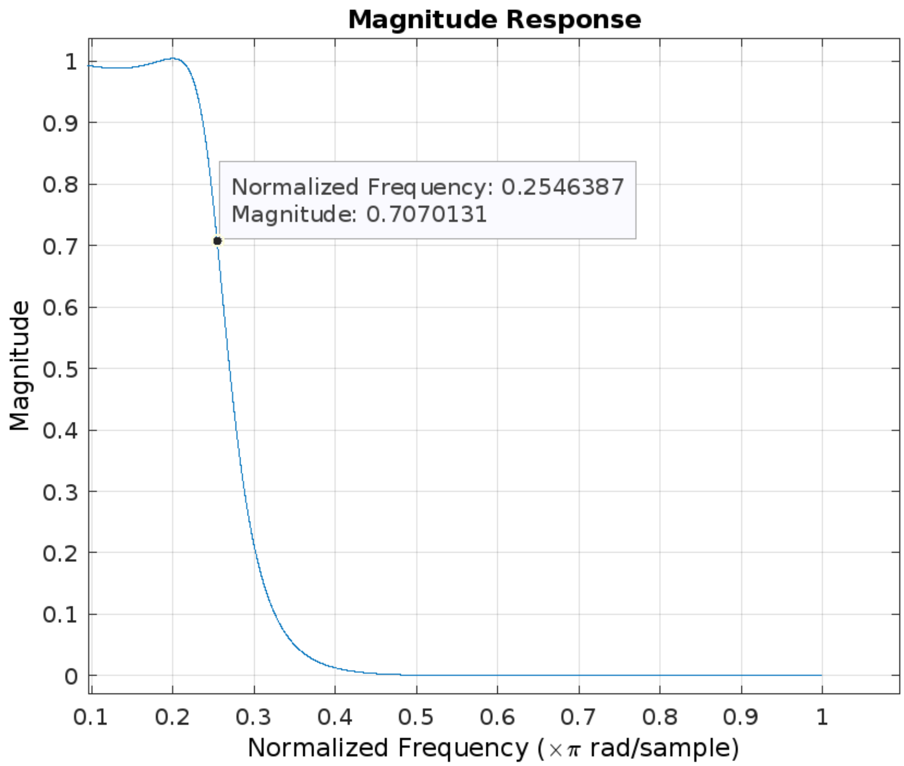
i) Is this filter IIR or FIR?

This is a IIR Filter.

iI) Is this filter Lowpass, Highpass, or Bandstop?

This is a low-pass filter.

iii) What is the approximate cutoff frequency of the filter?



Normalized Cutoff Frequency = 0.25464 $\left(\frac{1}{\pi}\right.$ rad/sample)

Cutoff Frequency = (0.25464) * (π) $\approx$ 0.8 rad/sample

Section 5 | part c)

```
n = 1:1:100;
y = cos(4*n) + cos(n/4);
y_high = cos(4*n);
y_low = cos(n/4);

y_filtered = filter(num, denom, y);

figure1 = figure();

subplot(2,2,1);
stem(y);
ylim([-2, 2]);
title("Origional Input Signal");
text(10, 1.5, "$$x[n]=cos(\frac{n}{4})*cos(4*n)$$", Interpreter="latex");

subplot(2,2,2);
stem(y_high);
```

```
ylim([-2, 2]);
title("High Frequency Component of Input Signal");
text(10, 1.5, "$$x[n]=cos(4*n)$$", Interpreter="latex");

subplot(2,2,3);
stem(y_low);
ylim([-2, 2]);
title("Low Frequency Component of Input Signal");
text(10, 1.5, "$$x[n]=cos(\frac{n}{4})$$", Interpreter="latex");

subplot(2,2,4);
stem(y_filtered);
ylim([-2, 2]);
title("Output of Filtered Signal");
```